# *16-782 Fall'24*
# *Planning & Decision-making in Robotics*

## *Introduction;*
## *What is Planning, Role of Planning in Robots*

*Maxim Likhachev*
*Robotics Institute*
*Carnegie Mellon University*

# Class Logistics

- Instructor:

  Maxim Likhachev – maxim@cs.cmu.edu

- TA:

  Itamar Mishani – imishani@andrew.cmu.edu
  Yorai Shaoul – yshaoul@andrew.cmu.edu

- Website:

  http://www.cs.cmu.edu/~maxim/classes/robotplanning_grad

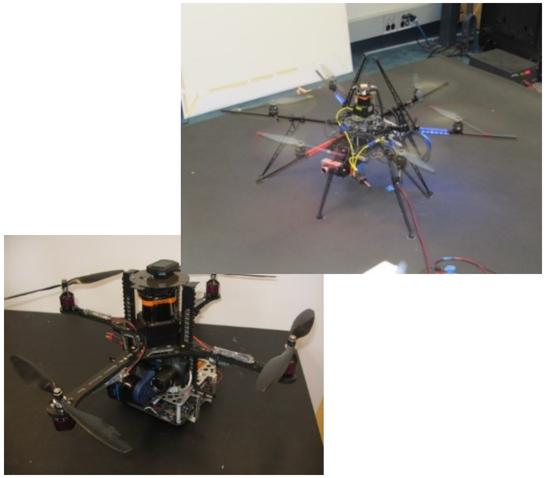
- Announcements, Questions, Recorded Lectures:

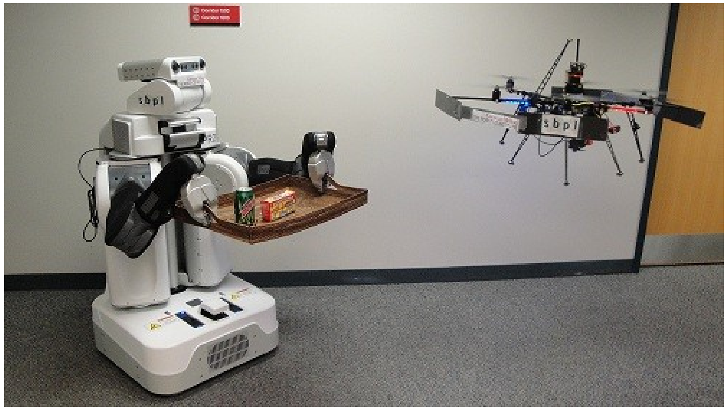
  - on Piazza
  - should have received an email with access info

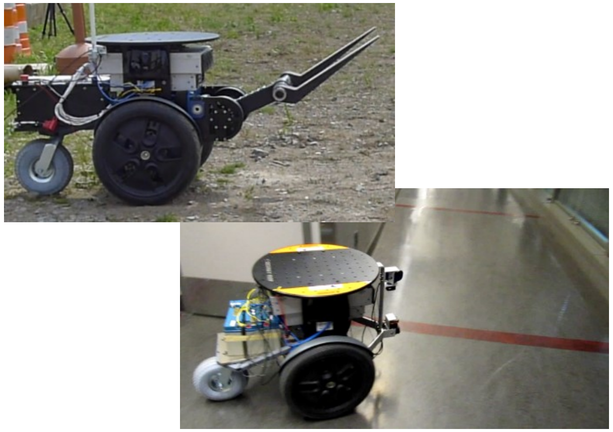
# About Me

- My Research Interests:
  - Planning, Decision-making, Learning
  - Applications: planning for complex robotic systems including aerial and ground robots, manipulation platforms, small teams of heterogeneous robots

- More info: [*http://www.cs.cmu.edu/~maxim*](http://www.cs.cmu.edu/~maxim)
- Search-based Planning Lab: [*http://www.sbpl.net*](http://www.sbpl.net)







# About Me

- Also, currently split between CMU and Waymo, where I'm heavily involved in planning for self-driving vehicles

# Class Objectives at High-level

- Understand and learn how to implement most popular planning and decision-making approaches in robotics
- Understand the challenges and basic approaches to interleaving planning and execution in robotic systems
- Learn common uses of planning/decision-making in robotics
- Get a sense for doing research in the area of planning/decision-making in robotics

# What is Planning?

- According to Wikipedia: *“Planning is the process of thinking about an organizing the activities required to achieve a desired goal.”*

# What is Planning **for Robotics**?

- According to Wikipedia: *"Planning is the process of thinking about an organizing the activities required to achieve a desired goal."*

- ***Given***
  - – *model (states and actions) of the robot(s)* $M^R = <S^R, A^R>$
  - – *a model of the world* $M^W$
  - – *current state of the robot* $s^R_{current}$
  - – *current state of the world* $s^W_{current}$
  - – *cost function C of robot actions*
  - – *desired set of states for robot and world G*

- ***Compute a plan π that***
  - – *prescribes a set of actions* $a_1,\ldots a_K$ *in* $A^R$ *the robot should execute*
  - – *reaches one of the desired states in G*
  - – *(preferably) minimizes the cumulative cost of executing actions* $a_1,\ldots a_K$

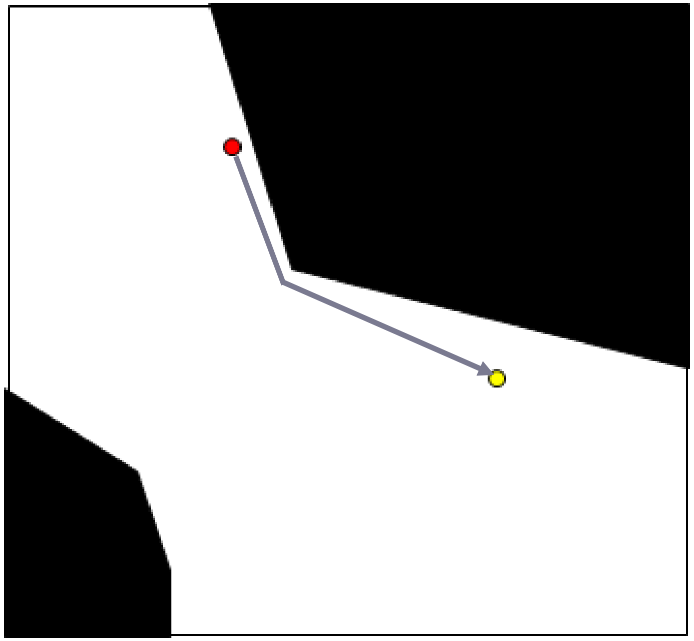
# Few Examples

- ***Given***
  - *– model (states and actions) of the robot(s)* $M^R = <S^R, A^R>$
  - *– a model of the world* $M^W$
  - *– current state of the robot* $s^R_{current}$
  - *– current state of the world* $s^W_{current}$
  - *– cost function C of robot actions*
  - *– desired set of states for robot and world G*

- ***Compute a plan π that***
  - *– prescribes a set of actions* $a_1, \dots a_K$ *in* $A^R$ *the robot should execute*
  - *– reaches one of the desired states in G*
  - *– (preferably) minimizes the cumulative cost of executing actions* $a_1, \dots a_K$

Planning for omnidirectional robot:

*What is* $M^R$*?*
*What is* $M^W$*?*
*What is* $s^R_{current}$*?*
*What is* $s^W_{current}$*?*
*What is C?*
*What is G?*



# Few Examples

- ***Given***
  - *– model (states and actions) of the robot(s)* $M^R = <S^R, A^R>$
  - *– a model of the world* $M^W$
  - *– current state of the robot* $s^R_{current}$
  - *– current state of the world* $s^W_{current}$
  - *– cost function C of robot actions*
  - *– desired set of states for robot and world G*

- ***Compute a plan π that***
  - *– prescribes a set of actions* $a_1, \ldots a_K$ *in* $A^R$ *the robot should execute*
  - *– reaches one of the desired states in G*
  - *– (preferably) minimizes the cumulative cost of executing actions* $a_1, \ldots a_K$

Planning for omnidirectional drone:

*What is* $M^R$*?*
*What is* $M^W$*?*
*What is* $s^R_{current}$*?*
*What is* $s^W_{current}$*?*
*What is C?*
*What is G?*

*MacAllister et al., 2013*

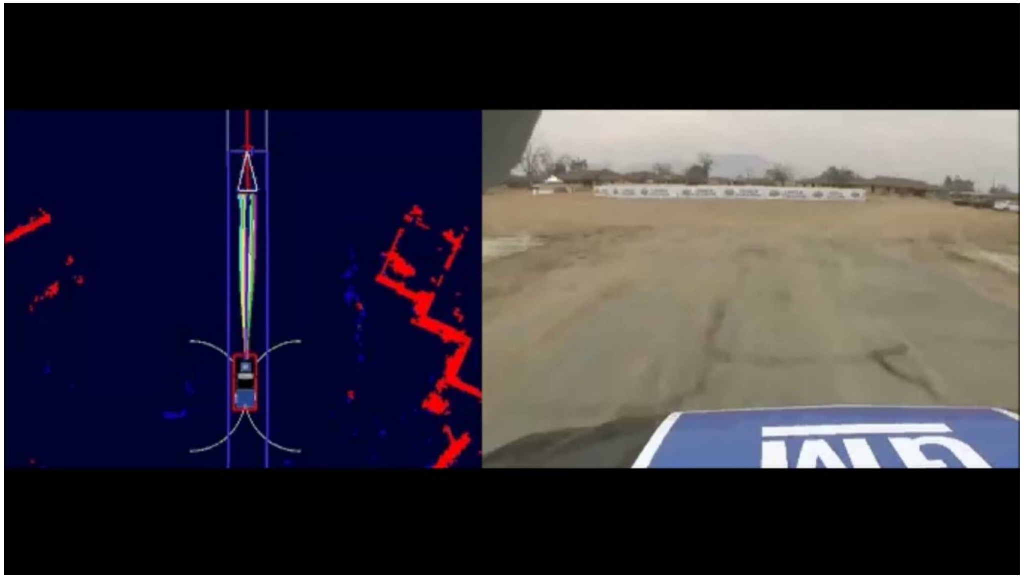
# Few Examples

- ***Given***
  - *model (states and actions) of the robot(s)* $M^R = <S^R, A^R>$
  - *a model of the world* $M^W$
  - *current state of the robot* $s^R_{current}$
  - *current state of the world* $s^W_{current}$
  - *cost function C of robot actions*
  - *desired set of states for robot and world G*

- ***Compute a plan π that***
  - *prescribes a set of actions* $a_1,\ldots a_K$ *in* $A^R$ *the robot should execute*
  - *reaches one of the desired states in G*
  - *(preferably) minimizes the cumulative cost of executing actions* $a_1,\ldots a_K$

Planning for autonomous navigation:

*What is* $M^R$*?*
*What is* $M^W$*?*
*What is* $s^R_{current}$*?*
*What is* $s^W_{current}$*?*
*What is C?*
*What is G?*



*Likhachev & Ferguson, '09; part of Tartanracing team from CMU for the Urban Challenge 2007 race*

# Few Examples

- ***Given***
  - *model (states and actions) of the robot(s)* $M^R = <S^R, A^R>$
  - *a model of the world* $M^W$
  - *current state of the robot* $s^R_{current}$
  - *current state of the world* $s^W_{current}$
  - *cost function C of robot actions*
  - *desired set of states for robot and world G*

- ***Compute a plan π that***
  - *prescribes a set of actions* $a_1,\ldots a_K$ *in* $A^R$ *the robot should execute*
  - *reaches one of the desired states in G*
  - *(preferably) minimizes the cumulative cost of executing actions* $a_1,\ldots a_K$

Planning for autonomous flight among people :

*Narayanan et al., 2012*

*What is* $M^R$*?*
*What is* $M^W$*?*
*What is* $s^R_{current}$*?*
*What is* $s^W_{current}$*?*
*What is C?*
*What is G?*

# Few Examples

- ***Given***
  - *– model (states and actions) of the robot(s)* $M^R = <S^R, A^R>$
  - *– a model of the world* $M^W$
  - *– current state of the robot* $s^R_{current}$
  - *– current state of the world* $s^W_{current}$
  - *– cost function C of robot actions*
  - *– desired set of states for robot and world G*

- ***Compute a plan π that***
  - *– prescribes a set of actions* $a_1,\ldots a_K$ *in* $A^R$ *the robot should execute*
  - *– reaches one of the desired states in G*
  - *– (preferably) minimizes the cumulative cost of executing actions* $a_1,\ldots a_K$

Planning for a mobile manipulator robot opening a door:

*Gray et al., 2013*

*What is* $M^R$*?*
*What is* $M^W$*?*
*What is* $s^R_{current}$*?*
*What is* $s^W_{current}$*?*
*What is C?*
*What is G?*

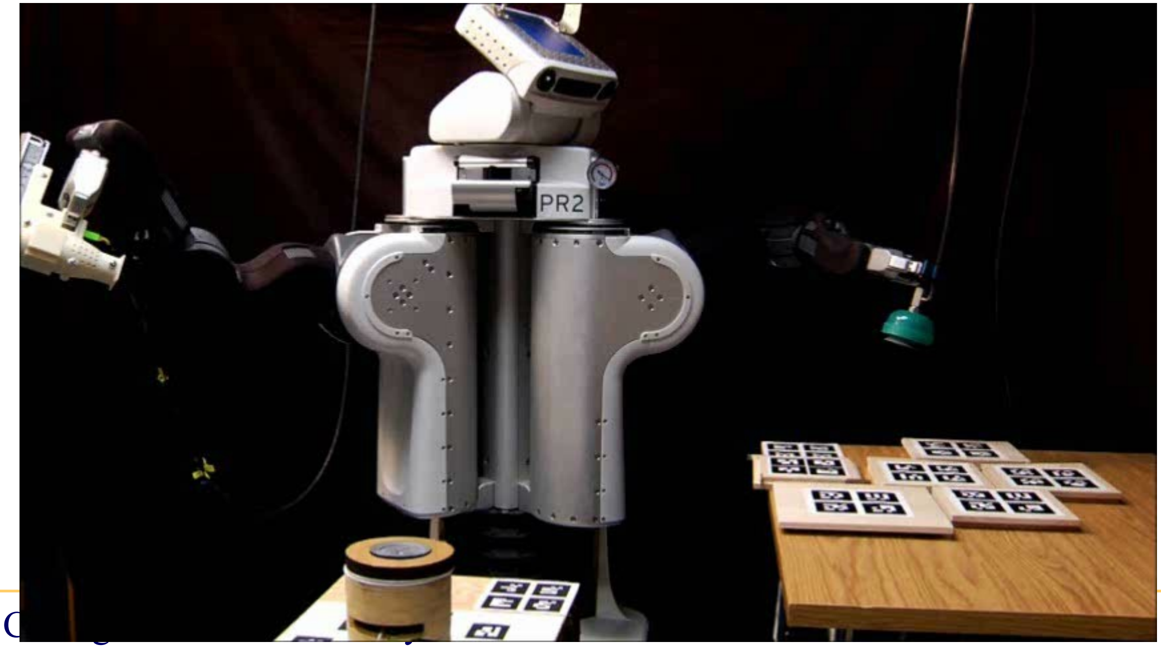
# Few Examples

- ***Given***
  - *– model (states and actions) of the robot(s)* $M^R = <S^R, A^R>$
  - *– a model of the world* $M^W$
  - *– current state of the robot* $s^R_{current}$
  - *– current state of the world* $s^W_{current}$
  - *– cost function C of robot actions*
  - *– desired set of states for robot and world G*

- ***Compute a plan π that***
  - *– prescribes a set of actions* $a_1, \ldots a_K$ *in* $A^R$ *the robot should execute*
  - *– reaches one of the desired states in G*
  - *– (preferably) minimizes the cumulative cost of executing actions* $a_1, \ldots a_K$

Planning for a mobile manipulator robot assembling a birdcage: *Cohen et al., 2015*

*What is* $M^R$*?*
*What is* $M^W$*?*
*What is* $s^R_{current}$*?*
*What is* $s^W_{current}$*?*
*What is C?*
*What is G?*



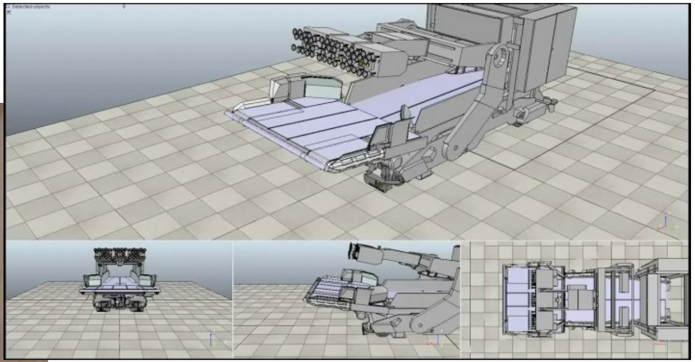
# Few Examples

- ***Given***
  - *– model (states and actions) of the robot(s)* $M^R = <S^R, A^R>$
  - *– a model of the world* $M^W$
  - *– current state of the robot* $s^R_{current}$
  - *– current state of the world* $s^W_{current}$
  - *– cost function C of robot actions*
  - *– desired set of states for robot and world G*

- ***Compute a plan π that***
  - *– prescribes a set of actions* $a_1, \ldots a_K$ *in* $A^R$ *the robot should execute*
  - *– reaches one of the desired states in G*
  - *– (preferably) minimizes the cumulative cost of executing actions* $a_1, \ldots a_K$

Planning/decision-making for a mobile manipulator unloading a truck:

*What is* $M^R$*?*
*What is* $M^W$*?*
*What is* $s^R_{current}$*?*
*What is* $s^W_{current}$*?*
*What is C?*
*What is G?*



# Assuming Infinite Computational Resources…

*Where does Planning break?*

# Assuming Infinite Computational Resources…

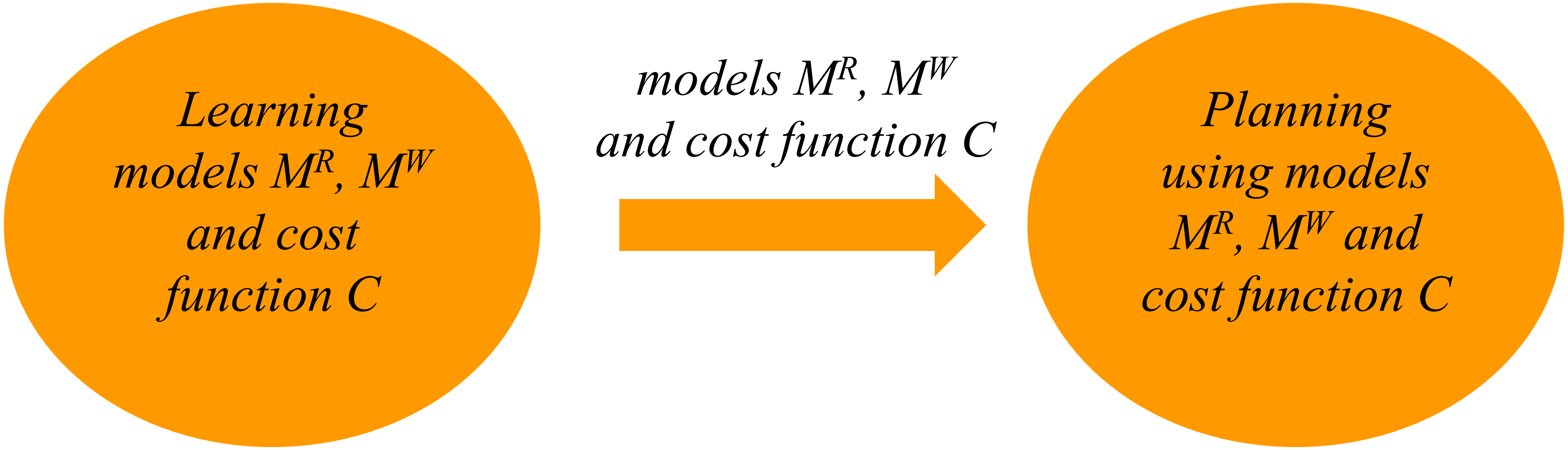
*Where does Planning break?*

*Reliance on the knowledge/accuracy of the model!*

*Role of Learning in Planning?*

# Planning vs. Learning

## *Model-based approach*



*Learning models $M^R$, $M^W$ and cost function $C$*

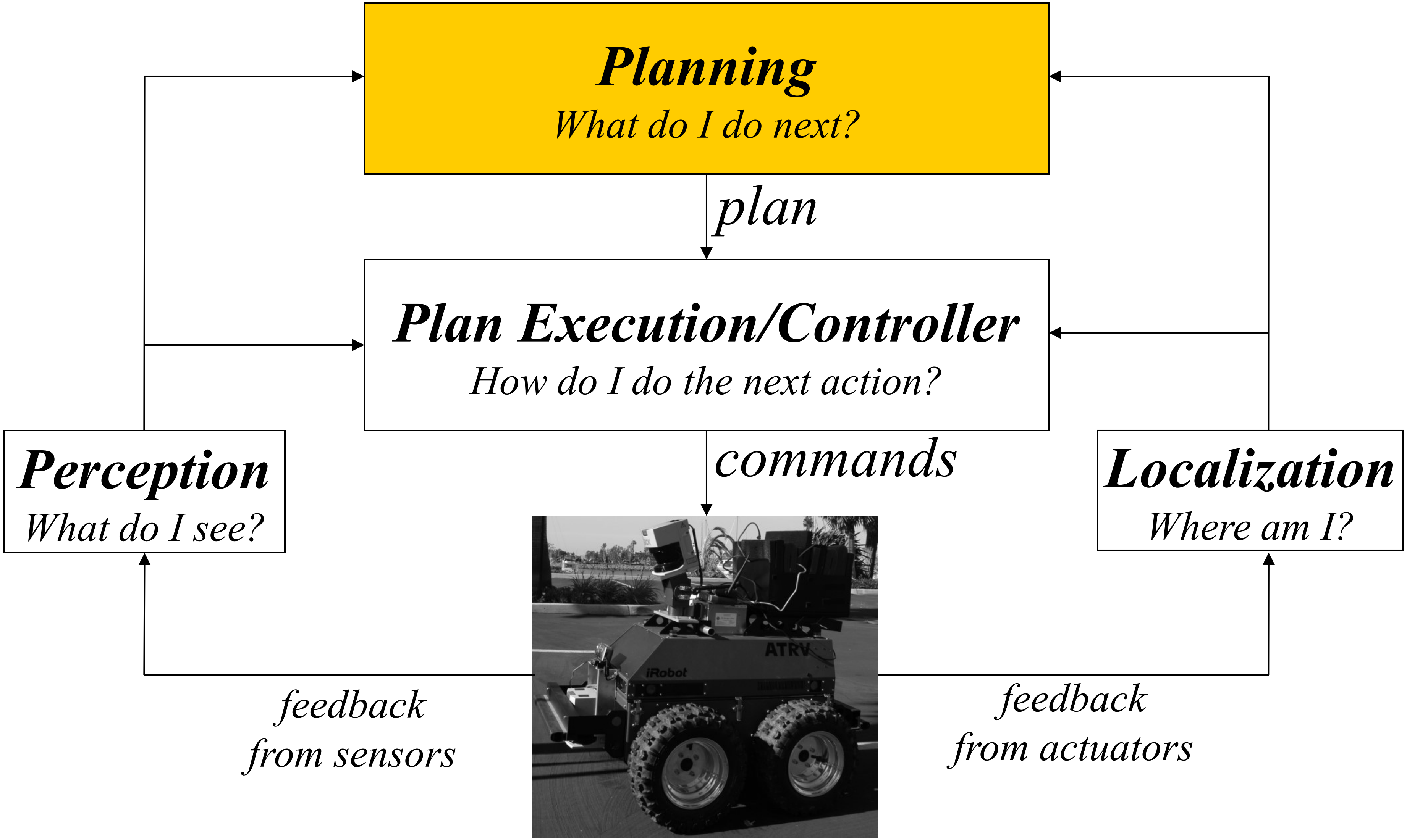
*models $M^R$, $M^W$ and cost function $C$*

*Planning using models $M^R$, $M^W$ and cost function $C$*

## *Model-free approach*

*Learning the mapping from "what robot sees" onto "what to do next" using rewards received by the robot (Reinforcement Learning) or demonstrations (Behavior Cloning)*

# Planning within a Typical Autonomy Architecture



**Planning**
*What do I do next?*

*plan*

**Plan Execution/Controller**
*How do I do the next action?*

*commands*

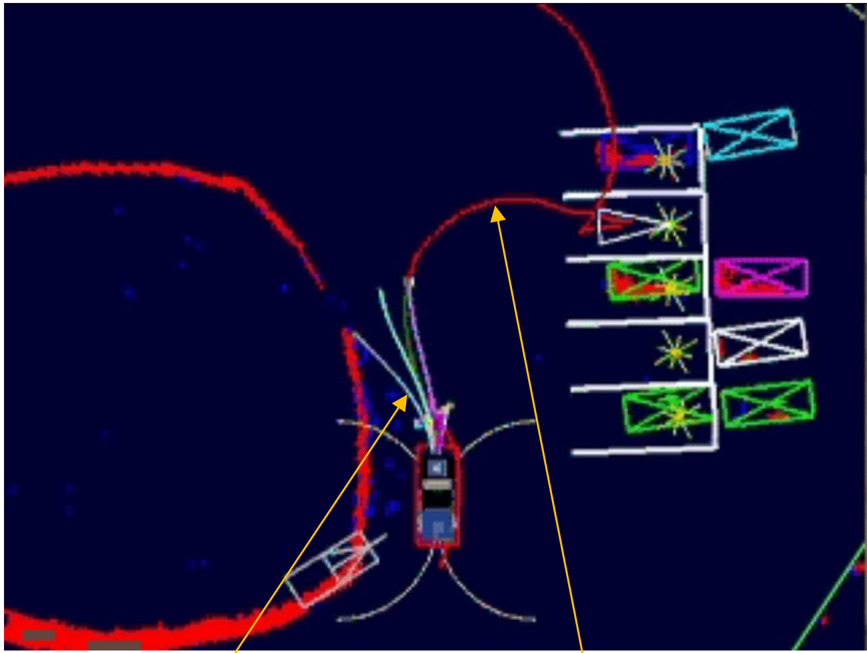
**Perception**
*What do I see?*

**Localization**
*Where am I?*

*feedback*
*from sensors*

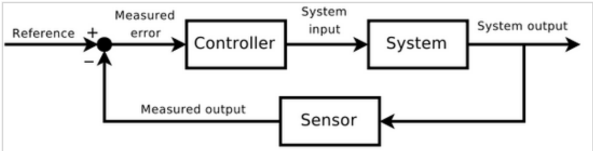
*feedback*
*from actuators*

# Planning vs. Trajectory Following vs. Control

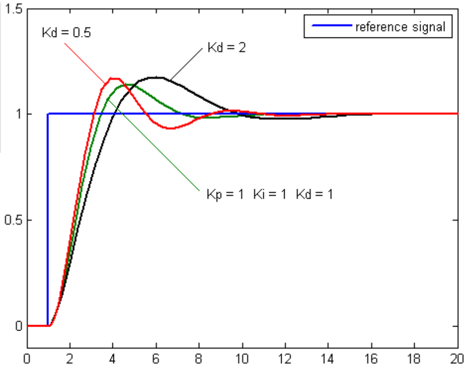


*local planning*
*(trajectory following)*

*global planning*



*controller*



*Images from wikipedia*

# Class Logistics

- Books (optional):

- Planning Algorithms *by Steven M. LaValle*

- Heuristic Search, Theory and Applications *by Stefan Edelkamp and Stefan Schroedl*

- Principles of Robot Motion, Theory, Algorithms, and Implementations *by Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A. Kantor, Wolfram Burgard, Lydia E. Kavraki and Sebastian Thrun*

- Artificial Intelligence: A Modern Approach *by Stuart Russell and Peter Norvig*

# Class Prerequisites

- Knowledge of programming (e.g., C, C++)

- Working knowledge of data structures & basic Computer Science algorithms (e.g., graphs, linked lists, priority queues, BFS/DFS, etc.)

- Prior exposure to robotics

# Class Objectives

- Understand and learn how to implement most popular planning algorithms in robotics including heuristic search-based planning algorithms, sampling-based planning algorithms, task planning, planning under uncertainty and multi-robot planning

- Learn basic principles behind the design of planning representations

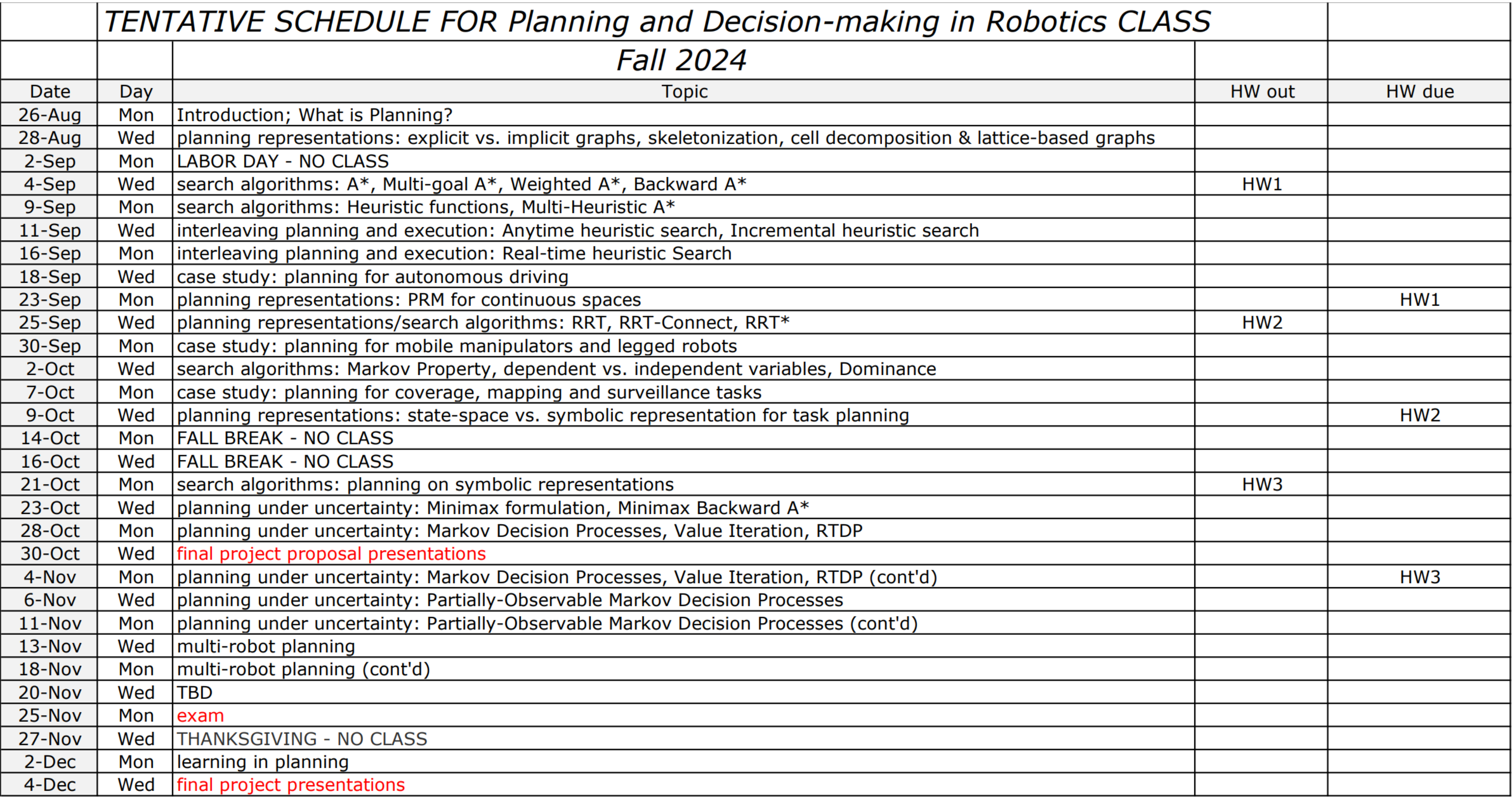
- Understand core theoretical principles that many planning algorithms rely on and learn how to analyze theoretical properties of the algorithms

- Understand the challenges and basic approaches to interleaving planning and execution in robotic systems

- Learn common uses of planning in robotics

- Get a sense for doing research in the area of planning/decision-making in robotics

# **Tentative** Class Schedule



*TENTATIVE SCHEDULE FOR Planning and Decision-making in Robotics CLASS*

*Fall 2024*

| Date | Day | Topic | HW out | HW due |
|---|---|---|---|---|
| 26-Aug | Mon | Introduction; What is Planning? | | |
| 28-Aug | Wed | planning representations: explicit vs. implicit graphs, skeletonization, cell decomposition & lattice-based graphs | | |
| 2-Sep | Mon | LABOR DAY - NO CLASS | | |
| 4-Sep | Wed | search algorithms: A*, Multi-goal A*, Weighted A*, Backward A* | HW1 | |
| 9-Sep | Mon | search algorithms: Heuristic functions, Multi-Heuristic A* | | |
| 11-Sep | Wed | interleaving planning and execution: Anytime heuristic search, Incremental heuristic search | | |
| 16-Sep | Mon | interleaving planning and execution: Real-time heuristic Search | | |
| 18-Sep | Wed | case study: planning for autonomous driving | | |
| 23-Sep | Mon | planning representations: PRM for continuous spaces | | HW1 |
| 25-Sep | Wed | planning representations/search algorithms: RRT, RRT-Connect, RRT* | HW2 | |
| 30-Sep | Mon | case study: planning for mobile manipulators and legged robots | | |
| 2-Oct | Wed | search algorithms: Markov Property, dependent vs. independent variables, Dominance | | |
| 7-Oct | Mon | case study: planning for coverage, mapping and surveillance tasks | | |
| 9-Oct | Wed | planning representations: state-space vs. symbolic representation for task planning | | HW2 |
| 14-Oct | Mon | FALL BREAK - NO CLASS | | |
| 16-Oct | Wed | FALL BREAK - NO CLASS | | |
| 21-Oct | Mon | search algorithms: planning on symbolic representations | HW3 | |
| 23-Oct | Wed | planning under uncertainty: Minimax formulation, Minimax Backward A* | | |
| 28-Oct | Mon | planning under uncertainty: Markov Decision Processes, Value Iteration, RTDP | | |
| 30-Oct | Wed | final project proposal presentations | | |
| 4-Nov | Mon | planning under uncertainty: Markov Decision Processes, Value Iteration, RTDP (cont'd) | | HW3 |
| 6-Nov | Wed | planning under uncertainty: Partially-Observable Markov Decision Processes | | |
| 11-Nov | Mon | planning under uncertainty: Partially-Observable Markov Decision Processes (cont'd) | | |
| 13-Nov | Wed | multi-robot planning | | |
| 18-Nov | Mon | multi-robot planning (cont'd) | | |
| 20-Nov | Wed | TBD | | |
| 25-Nov | Mon | exam | | |
| 27-Nov | Wed | THANKSGIVING - NO CLASS | | |
| 2-Dec | Mon | learning in planning | | |
| 4-Dec | Wed | final project presentations | | |

# Three Homeworks + Final Project

- <u>All homeworks are individual</u> (no groups)

- Final projects is a group project (3-5 people per group)

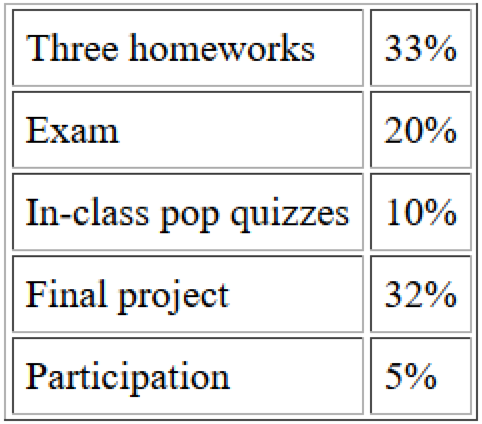
- Homeworks are programming assignments based on the material

- Final project is a research-like project
  - For example: to develop and implement a planner for a robot planning problem of your choice
  - Or: to extend a particular planning algorithm to improve its running time or to handle additional conditions

  - Two presentations (proposal and final) and meetings with groups

# Class Structure

- Grading



| | |
|---|---|
| Three homeworks | 33% |
| Exam | 20% |
| In-class pop quizzes | 10% |
| Final project | 32% |
| Participation | 5% |

- Exam is tentatively scheduled for Nov. 25

- Late Policy
  - 3 free late days
  - No late days may be used for the final project!
  - Each additional late day will incur a 10% penalty

Questions about the class?